

**Philosophische** Fakultät III

Sprach-, Literatur- und Kulturwissenschaften

Institut für Information und Medien, Sprache und Kultur (I:IMSK)  
Lehrstuhl für Informationswissenschaft

Modulprüfung Computational Intelligence

Modul: INF-BA-M09

SS 2021

Leitung: Stefan Kerscher/ Prof. Dr. Bernd Ludwig

**Trainieren eines neuronalen Netzes zur Prädiktion der Trajektorien von Fußgängern**

Pascal Strobel

Matrikelnummer: 2106133

6. Semester Medieninformatik/ Informationswissenschaft

E-Mail: pascal.strobel@stud.uni-regensburg.de

Abgegeben am 15.09.2021

Inhalt

[1 Einleitung 3](#_Toc81841092)

[2 Methodik 4](#_Toc81841093)

[2.1 Problemstellung 4](#_Toc81841094)

[2.2 Datenvorverarbeitung 5](#_Toc81841095)

[2.2.1 Normalisieren der Daten 5](#_Toc81841096)

[2.2.2 Standardisieren der Daten 5](#_Toc81841097)

[2.2.3 Formatieren der Daten 5](#_Toc81841098)

[2.3 Rekurrentes neuronales Netz (RNN) 5](#_Toc81841099)

[3 Evaluation 6](#_Toc81841100)

[3.1 Metriken 6](#_Toc81841101)

[3.2 Resultate 6](#_Toc81841102)

[3.3 Fälle des Scheiterns des Ansatzes 7](#_Toc81841103)

[4 Diskussion 10](#_Toc81841104)

[5 Fazit 11](#_Toc81841105)

[Literaturverzeichnis 12](#_Toc81841106)

[Erklärung zur Urheberschaft 13](#_Toc81841107)

# Einleitung

Autonomes Fahren, selbst fahrende Roboter am Arbeitsplatz etc. gewinnen in der heutigen Zeit zunehmend an Bedeutung und Aufmerksamkeit.

* Grenzt Thema inhaltlich genau ein: „Die Arbeit beschäftigt sich mit XY“
* Inwiefern ist die Problemstellung für die Informationswissenschaft relevant
* Pedestrian Trajectory prediction nimmt immer mehr an Wichtigkeit zu
* Beispielbereiche: Human Surveillance, Socio-Robot navigation und vor allem autonomes Fahren
* Einzelnen Ziele dieser Arbeit
* Ziel der Arbeit ist das Vorstellen eines Lösungsansatzes für die Problemstellung, sowie eine
* Inhaltlicher Aufbau der Arbeit
* Zunächst wird die eigentliche Problemstellung erläutert
* Dann wird der vorgeschlagene Lösungsansatz vorgestellt, darunter Datenvorverarbeitungsschritte, die Wahl des Neuronalen Netztes (und seine Parameter), sowie benötigte mathematischen Konzepte dahinter
* Anschließend erfolgt eine Evaluation des vorgestellten Ansatzes anhand der zwei in der Literatur gängigen Metriken ADE und FDE und es wird aufgezeigt, in welchen Fällen von Fußgängerverhalten der Ansatz versagt
* Im Anschluss werden die Ergebnisse diskutiert und Limitierungen, sowie mögliche Verbesserungsvorschläge aufgezeigt.
* Ein kurzes Fazit über den vorgestellten Ansatz rundet die Arbeit ab.

# Methodik

In diesem Abschnitt wird zunächst die Problemstellung dargelegt und die informationswissenschaftliche Fragestellung abgeleitet. Anschließend wird ein möglicher Lösungsansatz präsentiert, welcher neben den verwendeten Vorverarbeitungstechniken der Daten und der Wahl des trainierten neuronalen Netzes und seiner Hyperparameter, auch die zugehörigen mathematischen Grundlagen erläutert.

## Problemstellung

Die Prädiktion der Trajektorien von Fußgängern entspricht der Vorhersage des zukünftigen Bewegungspfades basierend auf einer bestimmten Anzahl zuvor beobachteter Positionen. Die informationswissenschaftliche Fragestellung, die sich daraus ergibt, und für die diese Arbeit einen Lösungsansatz liefert, lässt sich demnach folgendermaßen formulieren:

„Wie kann anhand bekannter, vorheriger Positionen die zukünftige Trajektorie eines Fußgängers bestimmt werden?“

Zur Beantwortung der Frage soll auf Basis eines gegebenen Datensatzes, welcher Daten von sich bewegenden Fußgängern enthält, ein neuronales Netz trainiert und eine Regression durchgeführt werden. Jeder Datenpunkt im Datensatz besteht dabei aus einer Zeitreihe mit insgesamt 20 Zeitschritten und den dazugehörigen Werten für die aktuelle x- und y-Koordinate (in Metern) des Fußgängers. Die zeitlichen Intervalle zwischen zwei Zeitpunkten innerhalb eines Datenpunktes betragen durchgehend 400ms. Neben den Koordinaten sind keine weiteren kontextuellen Informationen über den Fußgänger oder seine Umgebung bekannt.

Das trainierte neuronale Netz soll anschließend dazu genutzt werden, mithilfe der Informationen aus den ersten acht Zeitpunkten eines Datenpunktes, die zukünftigen zwölf Positionen des Fußgängers vorherzusagen.

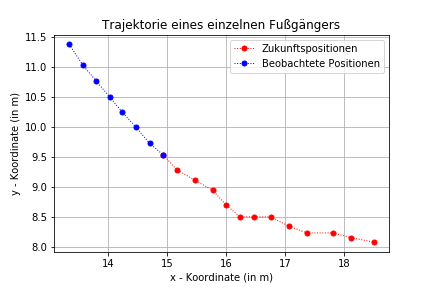


Abbildung : Beispiel-Trajektorie eines Fußgängers. Mithilfe der ersten acht Positionen (blau) sollen die zukünftigen zwölf (rot) vorhergesagt werden (eigene Darstellung)

Alle Grafiken mit [Matplotlib] erstellt 🡪 Fussnote machen

Diese acht-zu-zwölf-Verteilung ist gängig in der Literatur und beispielsweise in den Werken von [Social LSTM: Human Trajectory Prediction in Crowded Spaces | Pedestrian Trajectory Prediction with Convolutional Neural Networks | Convolutional Neural Network for Trajectory Prediction] zu finden. Die zugrunde liegende Aufgabe wird in Abb. 1 zusätzlich visualisiert.

## Datenvorverarbeitung

### Normalisieren der Daten

Der ursprüngliche Datensatz enthält die Positionen der Fußgänger als absolute Koordinaten. Da allerdings kein Ursprung für diese Koordinaten festgelegt ist, können die Bewegungsdaten unterschiedlicher Fußgänger absolut weit auseinander liegen (Fußgänger 1 z. B. im Bereich 1 bis 10m, Fußgänger 2 im Bereich 90 bis 100m).

[Pedestrian Trajectory Prediction with Convolutional Neural Networks] identifizieren neben dem naiven Ansatz, die absoluten Koordinaten ohne weitere Normalisierung zu verwenden, drei mögliche Normalisierungstechniken:

* Koordinaten haben den Ursprung im ersten beobachteten Zeitpunkt
* Koordinaten haben den Ursprung im letzten beobachteten Zeitpunkt
* Relative Koordinaten

Alle drei Ansätze führen dabei zu einer Performancesteigerung des in der Arbeit vorgestellten, faltenden neuronalen Netzes für unterschiedliche Datensätze, was die Notwendigkeit der Normalisierung der Daten nochmals betont.

In dieser Arbeit werden die Daten gemäß dem zuletzt aufgeführten Stichpunkt normalisiert. Das bedeutet, das das später trainierte neuronale Netz nur mit den relativen Änderungen der x- und y-Koordinaten zwischen zwei Zeitpunkten arbeitet, welche im weiteren Verlauf der Arbeit als Delta-x (dx) und Delta-y (dy) bezeichnet werden. Zur Bestimmung der Delta-Werte wurde für jeden Zeitpunkt eines Datenpunktes für alle Datenpunkte im Datensatz die Differenz der x- und y-Koordinaten vom aktuellen Zeitpunkt t und zum vorherigen Zeitpunkt t-1 berechnet. , formal dargestellt (evtl. Formel mit einfügen):

Das führt zu einer nötigen Spezialisierung der Anforderungen an das neuronale Netz: Aus den ersten acht Zeitschritten eines Datenpunktes werden je sieben dx- und dy-Werte berechnet. Das neuronale Netz soll demnach Zeitreihen der Länge sieben und den zwei Features dx und dy entgegennehmen und dafür die nächsten zwölf dx- und dy-Werte vorhersagen.

### Aufteilen der Daten in Trainings-, Validierungs- und Testdaten

Die normalisierten Datenpunkte wurden nach dem Zufallsprinzip in Trainings- (80%), Validierungs- und Testdaten (je 10%) aufgeteilt.

Zur Vermeidung von Data-Leakage erfolgte diese Aufsplittung vor der Standardisierung der Daten.

### Standardisieren der Daten

Um sicherzustellen, dass die Wertebereiche der zwei Merkmale dx und dy nicht zu unterschiedlich sind, werden diese mithilfe der z-Transformation standardisiert, sodass für jedes Merkmal ein Erwartungswert von 0 und eine Standardabweichung von 1 vorliegt. Der -Score des Samples lässt sich mit folgender der Formel berechnen:

ist dabei das arithmetische Mittel und die Standardabweichung der Trainingssamples.

Eine Verlagerung der Mittelwerte der Inputvariablen gegen null, wie es bei der z-Transformation der Fall ist, wird auch von [EfficientBackprop] empfohlen, da „any shift oft he average input away from zero will bias the updates in a particular direction and thus slow down learning“.

Die Standardisierung der Trainingsdaten, sowie die später folgende Umkehrung der z-Scores erfolgte mithilfe der StandardScaler- Klasse der Scikit-learn Bibliothek. (evtl. als Fußnote)

### Formatieren der Daten

In den richtigen Shape für das NN bringen + Aufteilen eines Datenpunktes in mehrere Samples

## Rekurrentes neuronales Netz (RNN)

RNN eignen sich besonders für Daten im Zeitreihenformat, allerdings gibt’s bei normalen RNN Probleme. Deshalb LSTM, welches auch gängig in der Literatur verwendet wird 🡪 allerdings sogar andere Ansätze (CNN), die besser abschneiden!

* Was ist LSTM/ CuDNNLSTM
* Was ist eine Dense Layer
* Was sind units
* Welche methoden wurden gegen Overfitting angewendet? 🡪 L1-Regularisierung 🡪 wie funktioniert das?
* Learning rate
* Epochen
* Adam optimizer
* Loss funktion

# Evaluation

## Metriken

Ebenso wie in verwandter Literatur von bspw. [Pedestrian Trajectory Prediction with Convolutional Neural Networks | Convolutional Neural Network for Trajectory Prediction] werden zur Auswertung der Performance die Metriken „Average Displacement Error“ (ADE), erstmalig eingeführt von [You’ll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking], und „Final Displacement Error“ (FDE) eingesetzt.

Der ADE eines Fußgängers ist die durchschnittliche euklidische Distanz zwischen den prädiktierten Positionen und der Ground-Truth für alle vorhergesagten Zeitschritte. Der ADE lässt sich mit folgender Formel berechnen:

ist der Zeitschritt der letzten Beobachtung des Fußgängers (in unserem Fall gilt = 8), der final zu vorhersagende Zeitschritt eines Datenpunktes (in unserem Fall gilt = 20). ist die vorhergesagte und die tatsächliche Position zum Zeitpunkt t. repräsentiert die euklidische Distanz.

Der FDE eines Fußgängers ist die euklidische Distanz zwischen der prädizierten Position und der zugehörigen Ground-Truth zum finalen Zeitpunkt und lässt sich folgendermaßen berechnen:

## Resultate

Die Performance des trainierten RNN wurde anhand des zu Beginn abgespaltenen Testdatensatzes (10% der Gesamtdaten), welcher die Trajektorien von insgesamt 540 Fußgängern umfasst, ausgewertet, indem für jeden Datenpunkt sowohl der ADE als auch der FDE bestimmt wurde. Damit eine gesammelte Aussage für die Performance des Systems gemacht werden kann, wurde der Durchschnitt der jeweiligen Metrik von alle 540 Datenpunkte berechnet, was zu folgendem Ergebnis führt:

Das RNN erzielt für die Testdaten einen durchschnittlichen ADE von 0,763 und einen durchschnittlichen FDE von 1,561.

Da x- und y-Koordinaten der Fußgänger in Metern angegeben sind, kann der ADE auch als der durchschnittliche Abstand in Metern zwischen einer vom System prädiktierten Zukunftsposition und der Grundwahrheit angesehen werden. Äquivalent dazu beschreibt der FDE den durchschnittlichen Abstand in Metern zwischen der prädiktierten und der tatsächlichen Position des Fußgängers zum finalen Zeitpunkt.

## Fälle des Scheiterns des Ansatzes

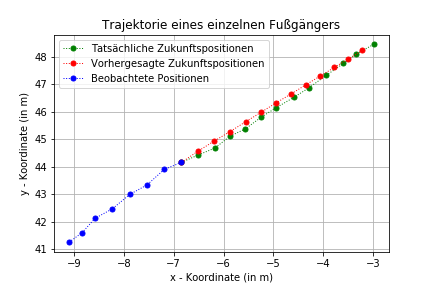


Abbildung : Präzise Vorhersagen durch triviales Fußgängerverhalten (eigene Darstellung)

Nur für wenige Fußgänger ist die Vorhersage der zukünftigen Trajektorie so trivial wie in Abb. 1. Abrupte Richtungswechsel oder Geschwindigkeitsänderungen sorgen für ein oft unberechenbares Verhalten von Fußgängern und damit zu Einbußen in der Performance. Unzuverlässige Prädiktionen des vorgestellten Systems sind dabei vor allem bei den folgenden vier Fällen (erstmal drei) zu beobachten.

1. Fußgänger ändert plötzlich Richtung oder Geschwindigkeit, ohne einen Hinweis darauf:

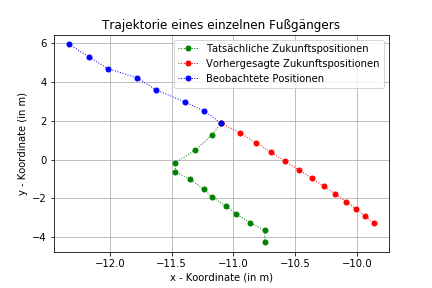


Abbildung 3: Überraschende Richtungsänderung (eigene Darstellung)

Die beobachtete Trajektorie des Fußgängers in Abb. 2 gibt keinerlei Hinweise auf eine mögliche Richtungsänderung. Die bestmögliche Vorhersage, die vom System (aber auch vom Menschen) getroffen werden kann, ist das Weiterführen dieser Bewegungsbahn. Damit derartige Bewegungsmuster besser prädiziert werden können, sind zusätzliche Kontext-Informationen (bspw. Straßenverlauf, Aufeinandertreffen mit anderen Fußgängern o.ä.) über den Fußgänger notwendig.

1. Fußgänger ändert mehrmals seine Richtung oder Geschwindigkeit:

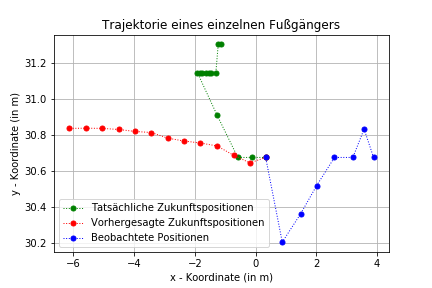


Abbildung 4: Mehrmalige Richtungs- und Geschwindigkeitsänderungen (eigene Darstellung)

Nicht nur plötzliche, unerwartete, sondern auch häufige Richtungs- und Geschwindigkeitsänderungen in den beobachteten Positionen, führen zu Schwierigkeiten bei der Vorhersage, wie in Abb. 3 ersichtlich. Ein richtiges Muster ist hierbei auch für den Menschen schwer erkennbar. Wie auch schon im ersten Fall, sind zusätzliche Kontext-Informationen nötig, die eine zuverlässigere Prädiktion erlauben.

1. Fußgänger ändert seine Position nicht:

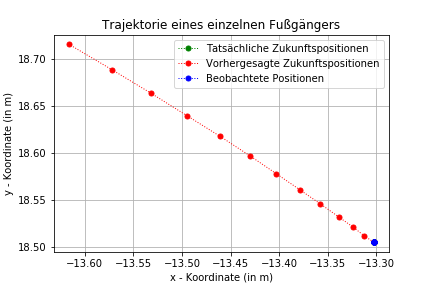


Abbildung 5: Keine Positionsänderung (eigene Darstellung)

Ebenfalls schlechte Vorhersagen des Systems sind zu beobachten, wenn sich der Fußgänger nicht bewegt, wie es bspw. in Abb. 4 der Fall ist. In derartigen Fällen scheint das System nicht gelernt zu haben, keinerlei Bewegung in x- und y- Richtung vorherzusagen. Stattdessen werden zukünftige Werte für dx und dy prädiziert, die leicht von Null abweichen. Durch das rekursive Vorgehen bei der Bestimmung der nächsten zwölf dx- und dy-Werte summiert sich dann der Fehler und führt zu zunehmenden Abweichungen zwischen Prädiktionen und Ground-Truth.

1. Fußgänger bewegt sich nach einem komplexeren Muster:

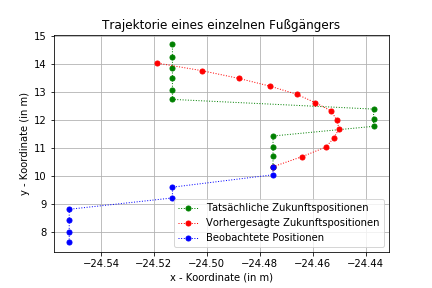


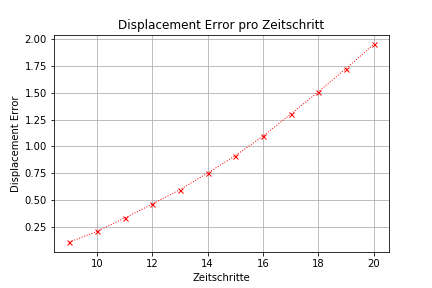
Abbildung 6: Komplexes Bewegungsmuster (eigene Darstellung)

* Bessere Abbildung finden, bei der der Mensch das Muster besser erkennen kann.
* Erst mal optional

Bei komplexeren Trajektorien

# Diskussion

Die

* Iteratives / Rekursives Predikten hat größeren Fehler zur Folge 🡪 siehe Grafik 🡪 besser wäre direkte Prediktion von 12 nächsten Pos
* Multi vs. Sequential Output. Trajectory prediction sequentially point-by-point performs poorly due to error propogation to future time-steps (trajectory curves off). Our multi-output model tends to be more resistant to such error accumulation. (aus dem zweiten Trajectory prediction paper with cnn) 
* Nur begrenzt Trainingsdaten
* Als Lösung: Data Augmentation (Rotieren, etc.)
* Hyperparameter wurden manuell optimiert / hyperparameter Tunin
* Nicht alle unterschiedlichen Kombinationen konnten getestet werden (z.B. wurde nur CudNNLSTM verwendet, da schneller, allerdings kann die activation Funktion nicht geändert werden, tanh steht sicher)
* Modell auf andere Datensätze (Trajnet etc.) anwenden, um mit bereits etablierten Ansätzen vergleichen zu können
* Auch komplett anderere Ansätze funktionieren, z.B. das Verwenden von CNN (siehe Paper) 🡪 vielleicht eher im Fazit

# Fazit

Diese Arbeit präsentiert einen möglichen Lösungsansatz zur Prädiktion von Fußgänger-Trajektorien mithilfe eines rekurrenten neuronalen Netzes. Dabei werden alle Datenvorverarbeitungsschritte, sowie das Training des neuronalen Netzes Der Ansatz liefert zufriedenstellende Ergebnisse für triviale Bewegungsbahnen

Trajektorie Predicition ist ein schwerer Task, was in diesem Paper auch nochmal klar wurde. Gerade aber für Anwendungen wie autonomes Fahren oder selbstfahrende Roboter am Arbeitsplatz, ist es enorm wichtig, sehr präzise Vorhersagen zu treffen, um Kollisionen mit Fußgängern zu verhindern.

Für triviale Vorhersagen ist das System zuverlässig, allerdings für die oben genannten Fehlerfälle nicht (da Fußgänger bewegung allerdings meistens unberechnbar sind, sind Änderungen am System, die eine präzisere Prädiktion, auch für nicht-triviale Cases, unbedingt notwendig)

Recursive Multi-Step Forecast

Literaturverzeichnis

Erklärung zur Urheberschaft

Wir haben die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie alle Zitate und Übernahmen von fremden Aussagen kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Die vorgelegten Druckexemplare und die vorgelegte digitale Version sind identisch.



Regensburg, 14.09.2021

|  |  |  |
| --- | --- | --- |
| Ort, Datum |  | Unterschrift |